Robotics and Autonomous Systems 70 (2015) 202–214

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Direction-dependent optimal path planning for autonomous vehicles

Alex Shum [a,*], Kirsten Morris [a], Amir Khajepour [b]

[a] Department of Applied Mathematics, University of Waterloo, 200 University Ave W, Waterloo, ON N2L 3G1, Canada

[b] Department of Mechanical and Mechatronics Engineering, University of Waterloo, 200 University Ave W, Waterloo, ON N2L 3G1, Canada

HIGHLIGHTS

- Optimal rover path planning is extended to consider direction in tip-over risk.
- Solar energy in net consumed energy is also considered in the optimal problem.
- The Ordered Upwind Method (OUM) is used to solve the rover path planning problem.
- A novel bi-directional (OUM) is introduced and is faster than the original algorithm.
- The bi-directional OUM is shown to outperform genetic algorithm and Bi-RRT*

ARTICLE INFO

Article history:
Received 5 February 2014
Received in revised form
3 December 2014
Accepted 4 February 2015
Available online 12 February 2015

Keywords:
Rover path planning
Ordered Upwind Method
Tip-over stability axes
Optimal path planning

ABSTRACT

The optimal path planning problem is considered for rovers. Tip-over risk is accurately modelled using direction dependence. In the previous direction-independent model, the value function was approximated using the Fast Marching Method (FMM). The risk was not accurately modelled. Solar energy is considered here for the first time. Minimizing path length, obstacle avoidance and soil risk are also considered. For a direction-dependent model, the value function in the optimal path planning problem can be approximated accurately using the Ordered Upwind Method (OUM) but not FMM. The value function is used to synthesize the optimal control, which is shown to have no local minima. A novel algorithmic improvement, OUM-BD over the OUM to include a bi-directional search is presented. The OUM-BD is slightly slower than the FMM, but can accurately solve a larger class of problems. The OUM-BD is faster than the existing OUM, an optimal bi-directional RRT path planner (Bi-RRT*), and a genetic algorithm (GA) path planner in terms of time, and outperforms both the GA and Bi-RRT* planner in cost in tested examples.

1. Introduction

Efficient and safe path planning is an important area of research for autonomous vehicles. The motivating application for the research described in this paper is path planning for rovers. However, the problem formulation and its solution can also be applied to path planning of other autonomous vehicles.

A path planning problem where the continuous environment is approximated by an undirected graph can be solved optimally using Dijkstra's [1] or A* [2] algorithms. The optimal path is found on the edges of the graph. Since the directions of travel are restricted to the edges, the optimal solution(s) of the discrete problem do not in general converge to the optimal solutions of the continuous problem even as the graph is refined [3].

Mixed-Integer Linear Programming (MILP) is often used to solve a discrete-time path planning problem for unmanned aerial vehicles (UAVs) [4–7]. In each time step, the time required to reach a goal position from its current position using a fixed number of waypoints is minimized. The dynamics of the vehicle are described using linear constraints. A set of linear and integer constraints are used to model both reaching the goal and polygonal obstacles. A cluttered environment for the MILP problem results in a significant increase in complexity. MILP is often used with receding horizon control to reduce computational effort [6,7]. The goal constraints are removed and replaced by a weight in the objective function. The resulting path is no longer globally optimal [4]. The path planning problem for rovers here places a strong emphasis on the properties of the terrain and is not easily described by the problem formulation of the MILP problem.

In rapidly-exploring random trees (RRTs) [8], collision-free connections are added to a growing tree data structure that stems from the start configuration. The goal configuration is added to the tree

* Corresponding author. Tel.: +1 519 888 4567x39244.
E-mail addresses: a5shum@uwaterloo.ca (A. Shum), kmorris@uwaterloo.ca (K. Morris), akhajepour@uwaterloo.ca (A. Khajepour).

http://dx.doi.org/10.1016/j.robot.2015.02.003
0921-8890/

when the growing tree reaches close enough to the goal configuration. A collision-free path is found using backtrack along the tree. A more efficient bi-directional algorithm called RRT-connect [9] grows a tree from each of the start and goal configuration, connecting the points of the two trees when they become close to one another. More recently, optimality for the RRT algorithm in RRT* [10] and its bi-directional variant Bi-RRT* [11] have been considered by rewiring existing connections on the tree, favouring connections with lower cost. Both RRT* and Bi-RRT* are asymptotically optimal, that is, the best path will converge to an optimal solution given an infinite number of iterations.

The continuous path planning problem can also be formulated as an optimal control problem [12], where aspects of the environment are modelled by a weight function. The value function according to the continuous dynamic programming principle (DPP) is first found. The optimal path is found by using the value function as a navigation function and solving an ODE related to the corresponding static Hamilton–Jacobi–Bellman (HJB) equation. Navigation functions may have local minima leaving the robot stuck without finding a feasible path [13]. Under a weak assumption on the weight function, the value function has no local minima [14]. Thus, the optimal path can always be recovered from the value function.

The static HJB equation reduces to the Eikonal equation for a direction-independent weight. In this case, the solution can be approximated using the Fast Marching Method (FMM) [15]. Such weights include obstacle avoidance and shortest path [16,17]. Weights for soil type and solar energy absorption as presented in this paper are also direction-independent. The solution is finalized one grid/mesh point at a time. The FMM applied to problems with weights that depend on direction will lead to residual error [12].

The Ordered Upwind Method (OUM) [12] was introduced to solve both a time-invariant continuous dynamic programming principle (DPP) problem, as well as a static Hamilton–Jacobi equation. In both these instances, the weight could have varied with not only position, but also direction. In the work here, the formulation regarding the DPP is considered. The OUM produces an approximation to the solution of the static Hamilton–Jacobi–Bellman equation [12] on an unstructured mesh. It has been shown [12] that the approximation produced by the OUM converge to the true solution of the static HJB equation as the mesh is refined. The characteristics to the solution of the Hamilton–Jacobi–Bellman equation are optimal paths given specific boundary conditions, and hence characteristics of the approximated solution from OUM are approximations of the optimal paths.

In this paper, the OUM is used to solve path planning problems for rovers. Contributions include a more accurate model of tip-over risk and the introduction of solar energy used in optimal path planning for rovers. Minimizing path length, obstacle avoidance and soil risk are also considered. A novel algorithmic improvement, OUM-BD, combining a bi-directional search using OUM is introduced. The OUM-BD is compared against OUM, FMM (for applicable problems), a genetic algorithm (GA) path planner for rovers [18], and Bi-RRT* [11] in both performance and timing. The OUM-BD is observed to be slightly slower than the FMM, but much faster than the original OUM. All of FMM, OUM-BD and OUM outperform GA and Bi-RRT* in timing and performance in all examples.

The path planning problem for rovers will be presented in Section 2. The optimal control problem, continuous DPP and static HJB are described in Section 3. In Section 4, a brief review of the Ordered Upwind Method will be presented. The OUM-BD algorithm to find the optimal path will be proposed in Section 5. Numerical examples will be presented in Section 6. Conclusions and extensions for future work are discussed in Section 7. Finally, the solution of the static HJB equation will be shown to have no local minima in the Appendix.

2. Problem statement

The environment, also known as the workspace, in which the path is planned is assumed to be fully known. For the rover path planning problem, it is assumed that the elevation of the terrain, position of the sun and locations of potential obstacles are all known. Let $|\cdot|$ denote the Euclidean norm.

Definition 2.1. The **workspace** $\Omega \subset \mathbb{R}^2$ is a closed, bounded and convex set of possible locations of the vehicle.

Let $\mathbf{x} : \mathbb{R}_+ \to \mathbb{R}^2$ describe the trajectory of the vehicle. For notational purposes, $\mathbf{x}(t)$ will denote the trajectory at time t and $\mathbf{x}$ will denote a point in $\mathbb{R}^2$. The direction in which the vehicle is facing is defined $\mathbf{u} \in \mathcal{U}$, where $\mathcal{U} = \{\mathbf{u}(\cdot) : \mathbb{R}_+ \to \mathbb{S}^1 | \mathbf{u}(\cdot) \text{ is measurable}\}$ and $\mathbb{S}^1 = \{\mathbf{u} \in \mathbb{R}^2 \big| |\mathbf{u}| = 1\}$. The dynamics of the rover for $t \geq 0$ are

$$\dot{\mathbf{x}}(t) = \mathbf{u}(t) \tag{1}$$
$$\mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}_0 \in \Omega.$$

Rovers have slow maximum speed (5 cm/s) and assume this speed quickly [19].

The control problem is to find a control $\mathbf{u}(\cdot) \in \mathcal{U}$ such that the vehicle will travel from a start position, $\mathbf{x}_0$ to a final position, $\mathbf{x}_f$ under the dynamics of the system (1).

Definition 2.2. The **exit-time** $T : \Omega \times \mathcal{U} \to \mathbb{R}_+$ is

$$T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \inf\{t \in \mathbb{R}_+ | \mathbf{x}(t) = \mathbf{x}_f\}.$$

The exit-time is the first time the vehicle reaches $\mathbf{x}_f$ (from $\mathbf{x}_0$) under the influence of $\mathbf{u}(\cdot)$.

Definition 2.3. The **cost function**

$$\text{Cost}(\mathbf{x}_0, \mathbf{u}(\cdot)) = \int_0^{T(\mathbf{x}_0, \mathbf{u}(\cdot))} g(\mathbf{x}(s), \mathbf{u}(s))ds, \tag{2}$$

where $\mathbf{x}(T(\mathbf{x}_0, \mathbf{u}(\cdot))) = \mathbf{x}_f$ and the **weight function** $g : \Omega \times \mathbb{S}^1 \to \mathbb{R}_+$ is a real-valued continuous function that satisfies

$$0 < G_{min} < g_{min}(\mathbf{x}) \leq g(\mathbf{x}, \mathbf{u}) \leq g_{max}(\mathbf{x}) < G_{max} < \infty,$$
$$\text{for every } (\mathbf{x}, \mathbf{u}) \in \Omega \times \mathbb{S}^1, \tag{3}$$

where $g_{min}(\mathbf{x}) = \min_{\mathbf{u} \in \mathbb{S}^1} g(\mathbf{x}, \mathbf{u})$ and $g_{max}(\mathbf{x}) = \max_{\mathbf{u} \in \mathbb{S}^1} g(\mathbf{x}, \mathbf{u})$ are continuous, and G_{min} and G_{max} are positive constants.

The assumption that there is a maximum bound on the weight G_{max} over Ω to traverse a point is equivalent to small-time local controllability (STLC) over all of Ω. States with infinite weight cannot be traversed and in general cannot be part of the solution to any path planning problem. This assumption is not reasonable in general for wheeled robots with a minimum turning radius. The STLC property is however justified for rovers which can turn in place. Since the problem formulation has no fixed exit-time, a negative weight may imply that the cost is not bounded below. Regions of weight zero can imply arbitrarily large exit-times (the rover remains in such regions without accruing additional cost). The assumption of a positive weight ensures that there are no positions that can be traversed for zero cost.

The weight function $g(\mathbf{x}, \mathbf{u}) \equiv 1$, for all $(\mathbf{x}, \mathbf{u}) \in \Omega \times \mathbb{S}^1$ corresponds to the optimal control problem for shortest time (and hence shortest path, since the vehicle travels at constant speed). The shortest path is not necessarily the optimal path.

The above formulation applies to path planning for any autonomous vehicle. Weightings specific to rovers are considered in this paper, including shortest path on terrain, obstacle avoidance, soil risk, solar energy input and tip-over stability risk. Some specific types of weights are as follows.